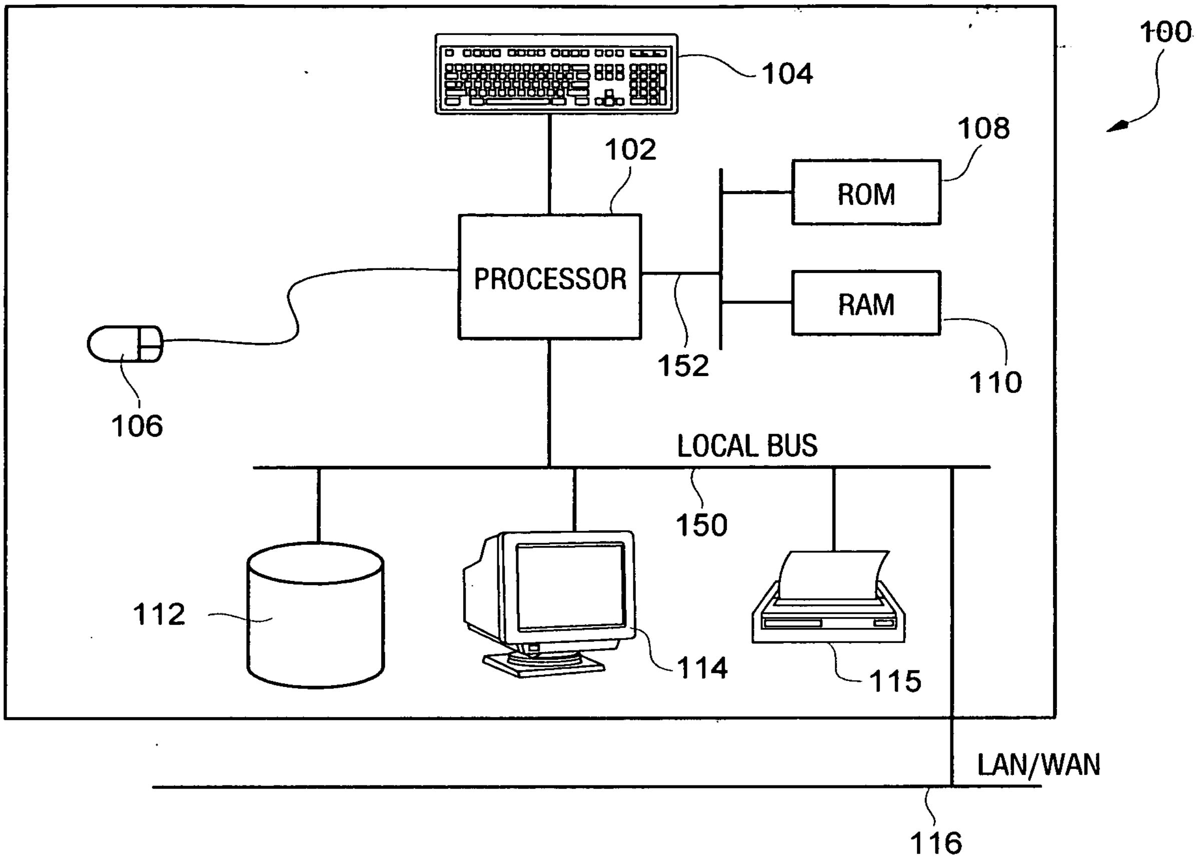
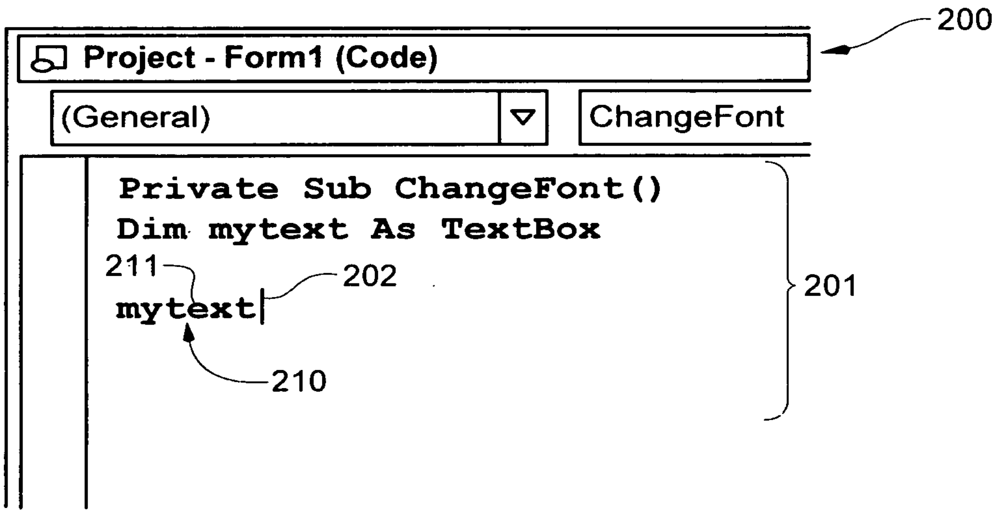
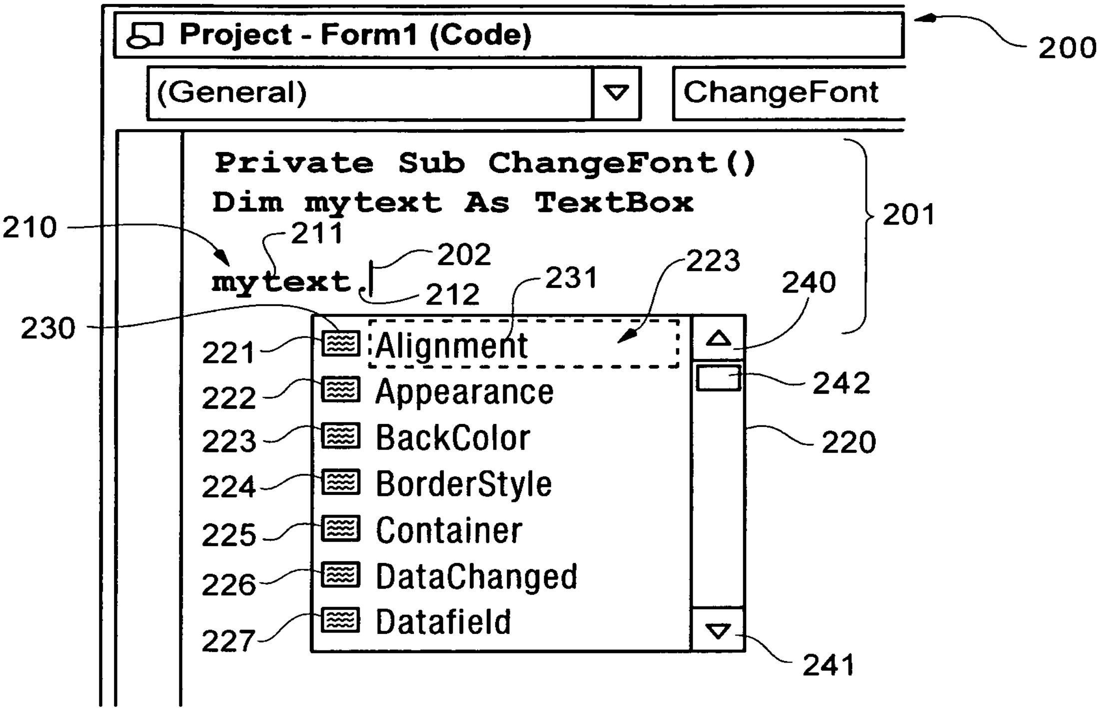


FIG. 1



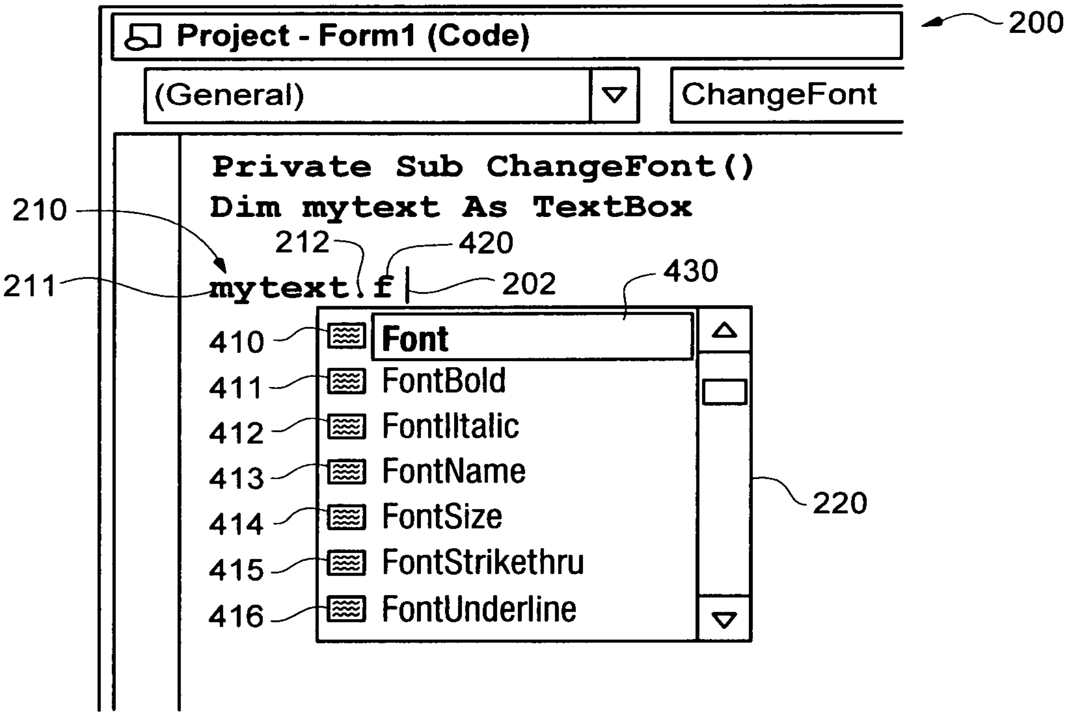
```
Private Sub ChangeFont()
Dim mytext As TextBox
mytext|
```

FIG. 2



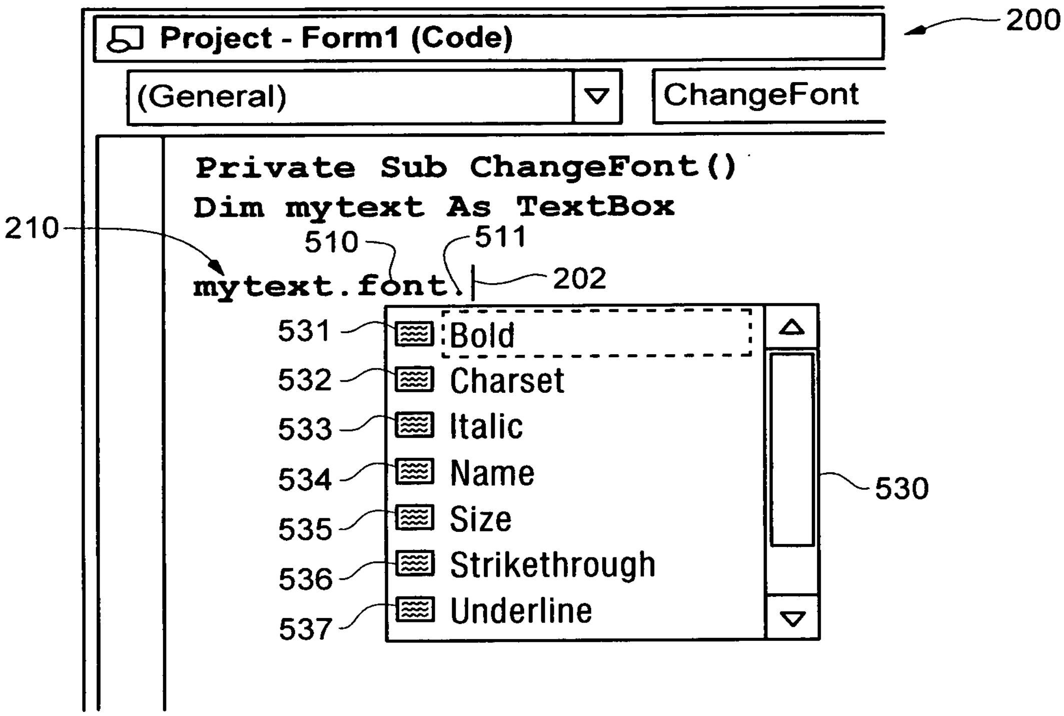
```
Private Sub ChangeFont()
Dim mytext As TextBox
mytext.
```

FIG. 3



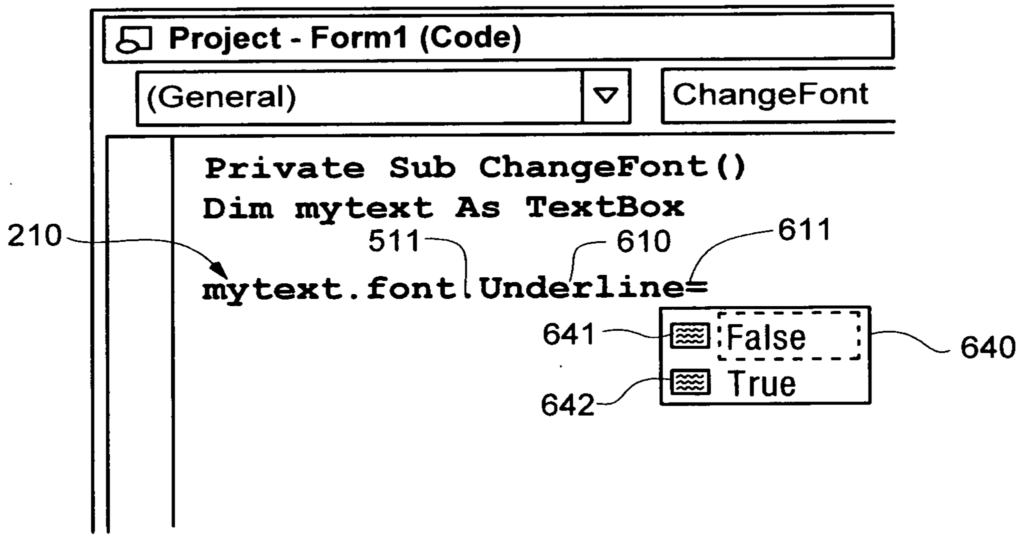
```
Private Sub ChangeFont()
Dim mytext As TextBox
mytext.f
```

FIG. 4



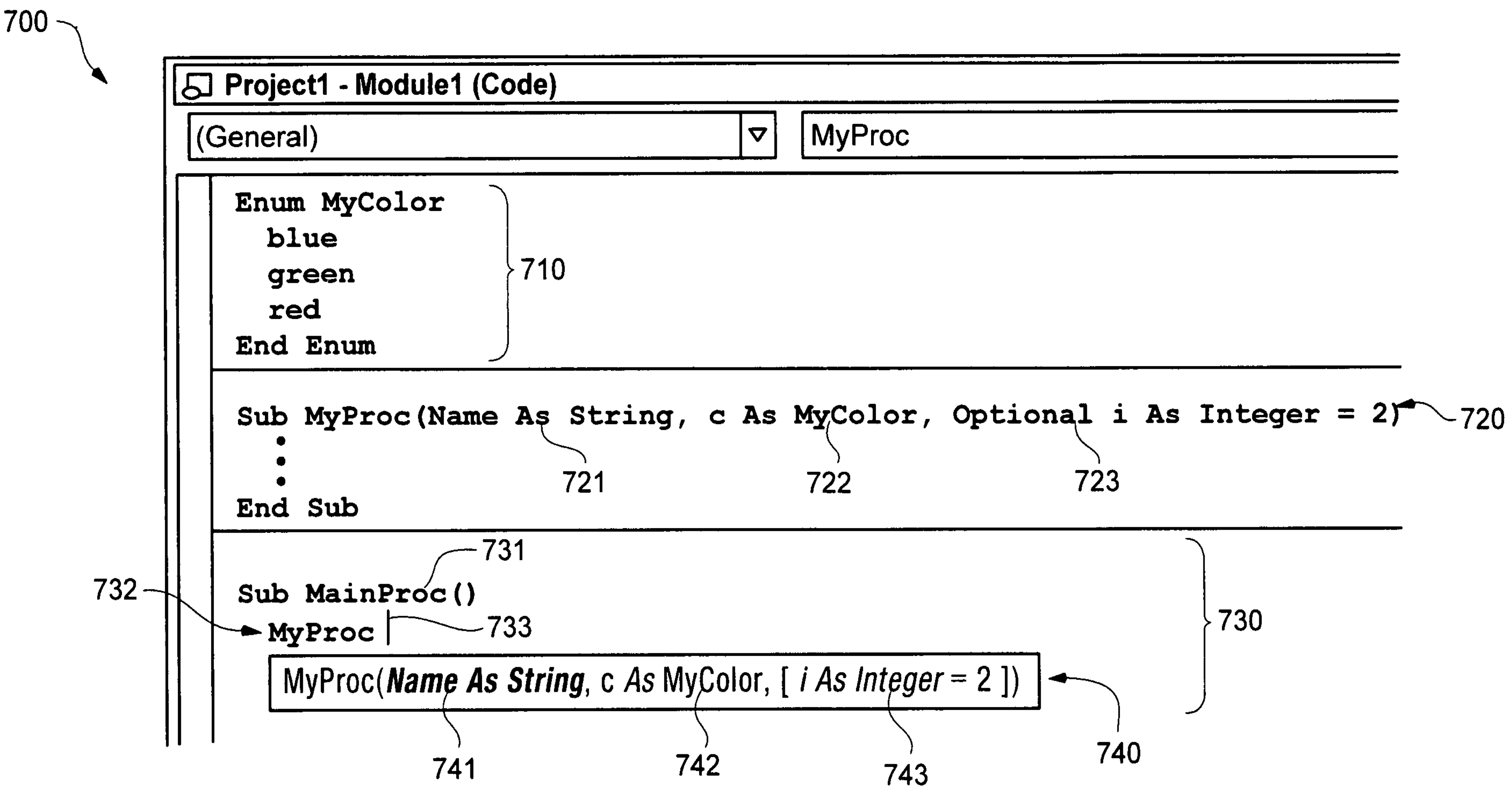
```
Private Sub ChangeFont()
Dim mytext As TextBox
mytext.font.
```

FIG. 5



```
Private Sub ChangeFont()
Dim mytext As TextBox
mytext.font.Underline=
```

FIG. 6

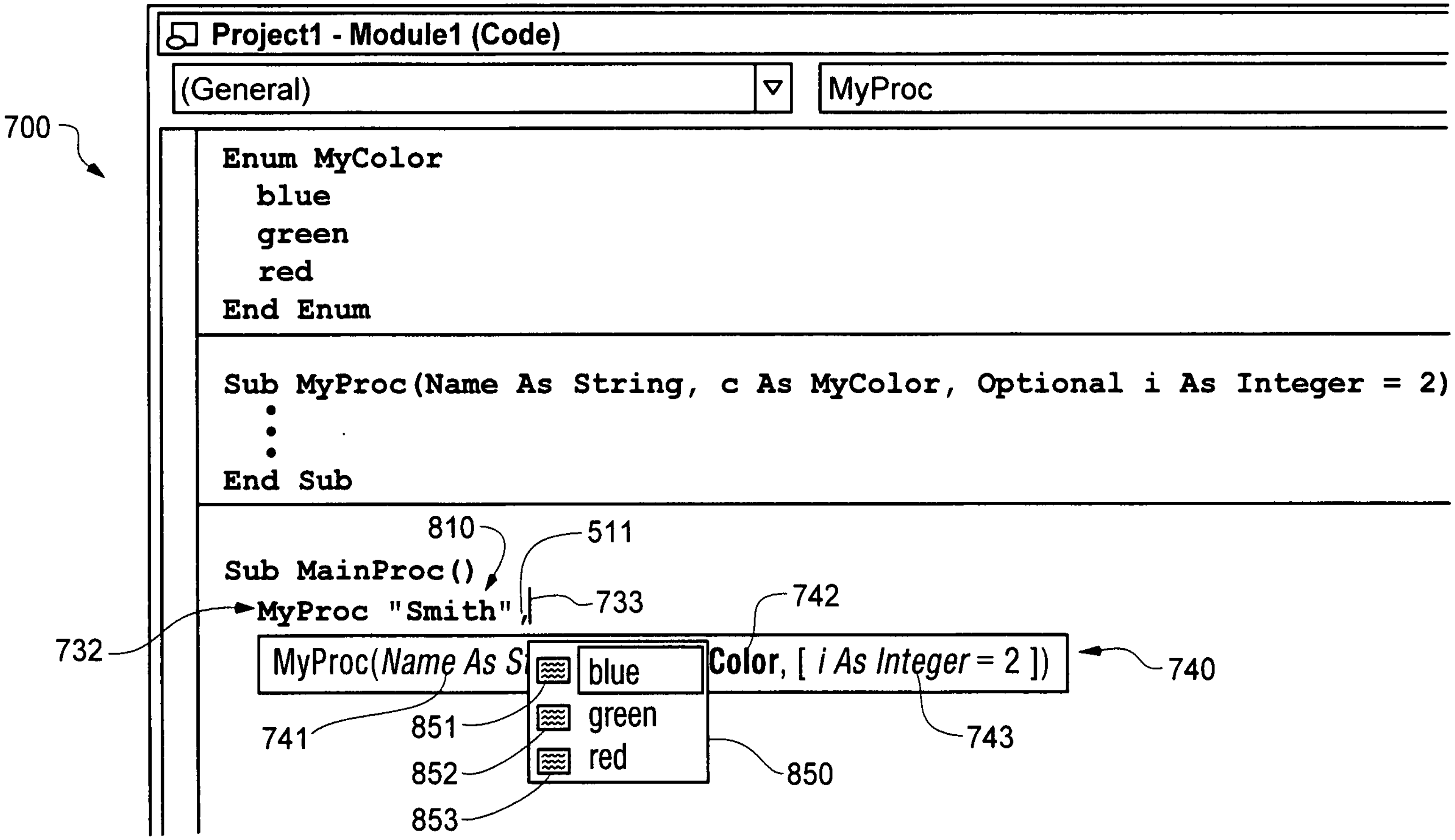


```
Enum MyColor
  blue
  green
  red
End Enum

Sub MyProc(Name As String, c As MyColor, Optional i As Integer = 2)
  •
  •
  •
End Sub

Sub MainProc()
  MyProc |
  MyProc(Name As String, c As MyColor, [ i As Integer = 2 ])
```

FIG. 7



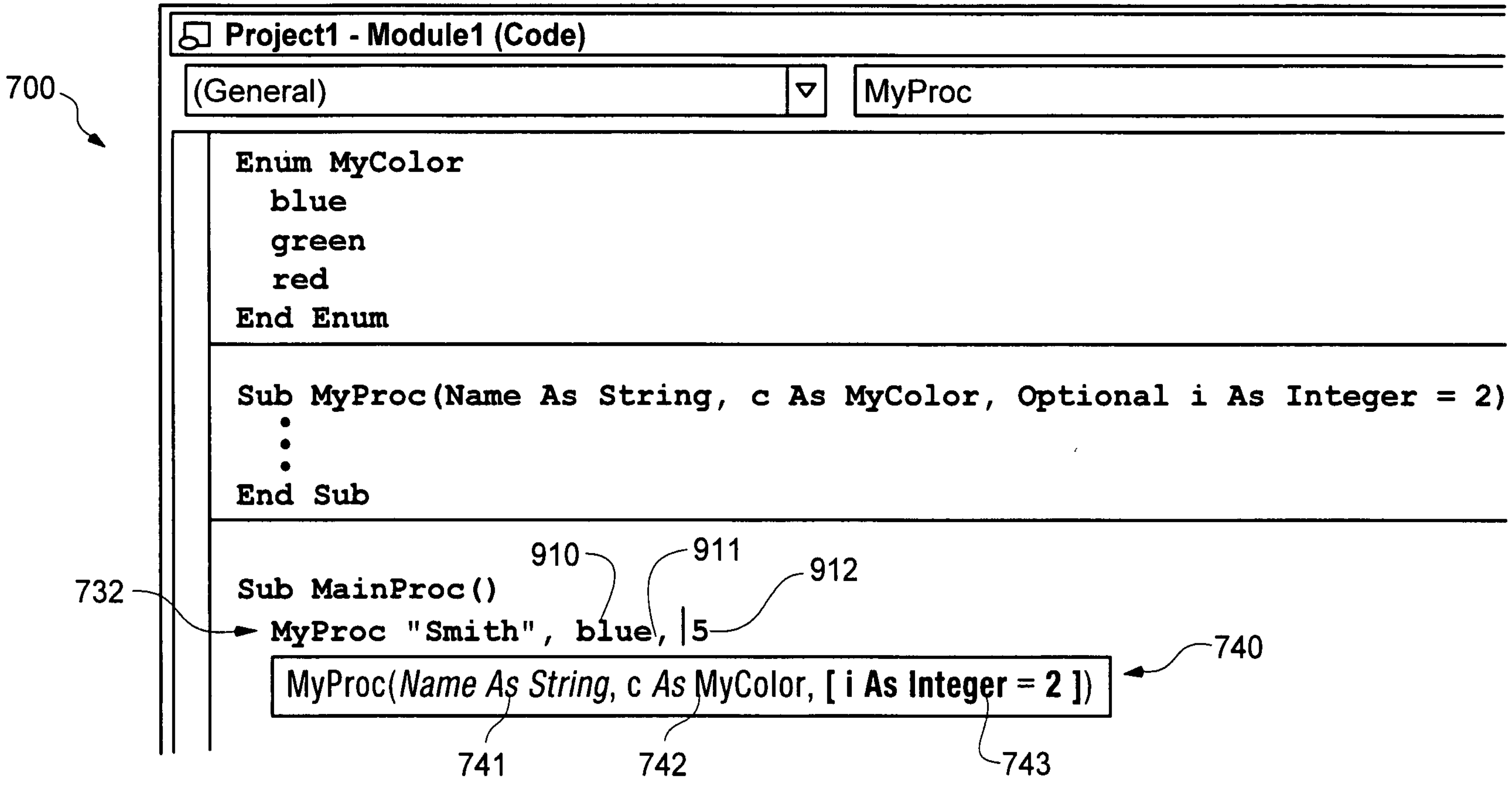
```
Enum MyColor
  blue
  green
  red
End Enum

Sub MyProc(Name As String, c As MyColor, Optional i As Integer = 2)
  •
  •
  •
End Sub

Sub MainProc()
  MyProc "Smith",|
```

MyProc(*Name As S*… **Color**, [*i As Integer* = 2])

FIG. 8



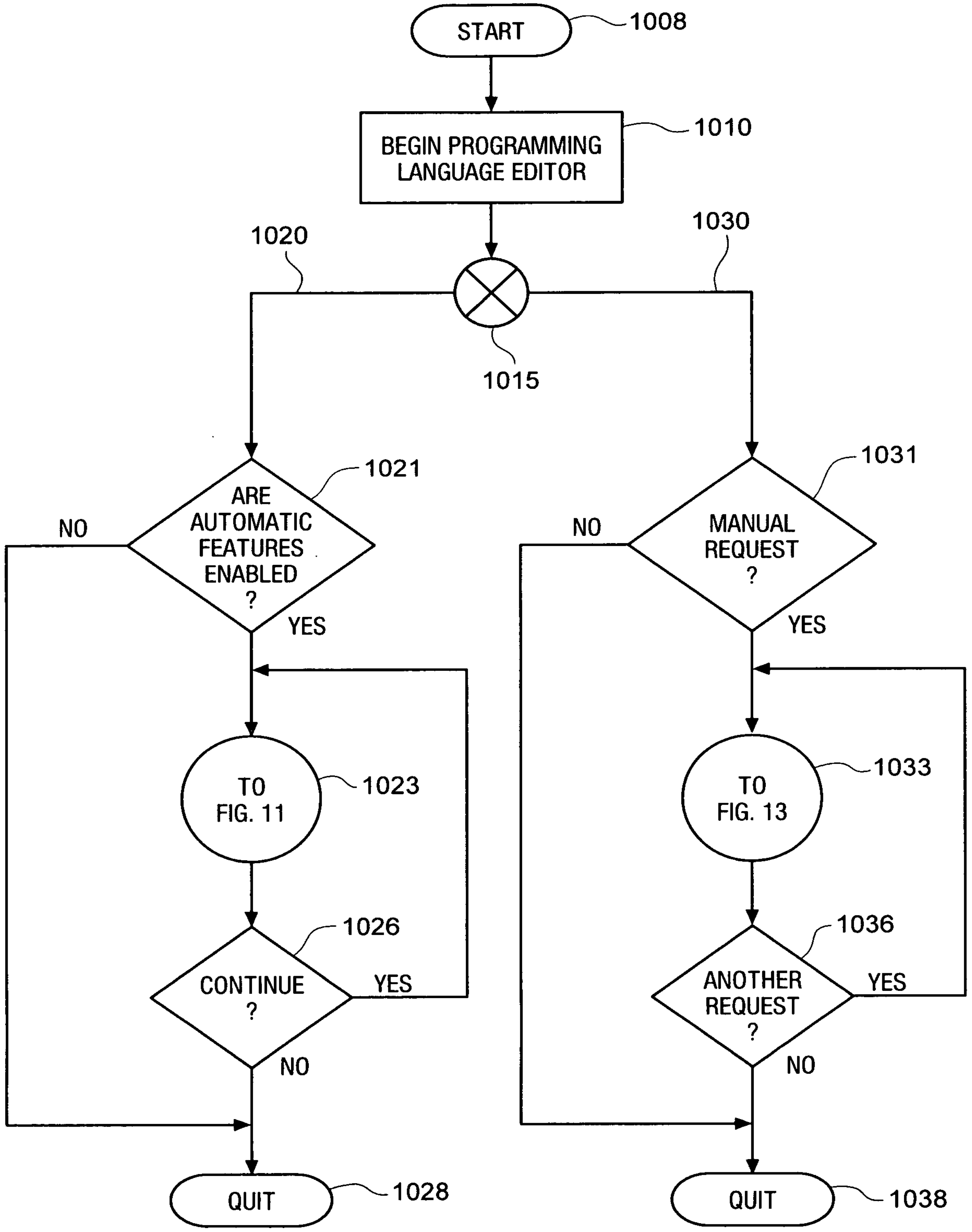
```
Enum MyColor
  blue
  green
  red
End Enum

Sub MyProc(Name As String, c As MyColor, Optional i As Integer = 2)
   •
   •
   •
End Sub

Sub MainProc()
  MyProc "Smith", blue, |5
  MyProc(Name As String, c As MyColor, [ i As Integer = 2 ])
```

FIG. 9

FIG. 10



START
1008
BEGIN PROGRAMMING LANGUAGE EDITOR
1010
1020
1030
1015
ARE AUTOMATIC FEATURES ENABLED ?
1021
NO
YES
TO FIG. 11
1023
CONTINUE ?
1026
YES
NO
QUIT
1028
MANUAL REQUEST ?
1031
NO
YES
TO FIG. 13
1033
ANOTHER REQUEST ?
1036
YES
NO
QUIT
1038

FIG. 11

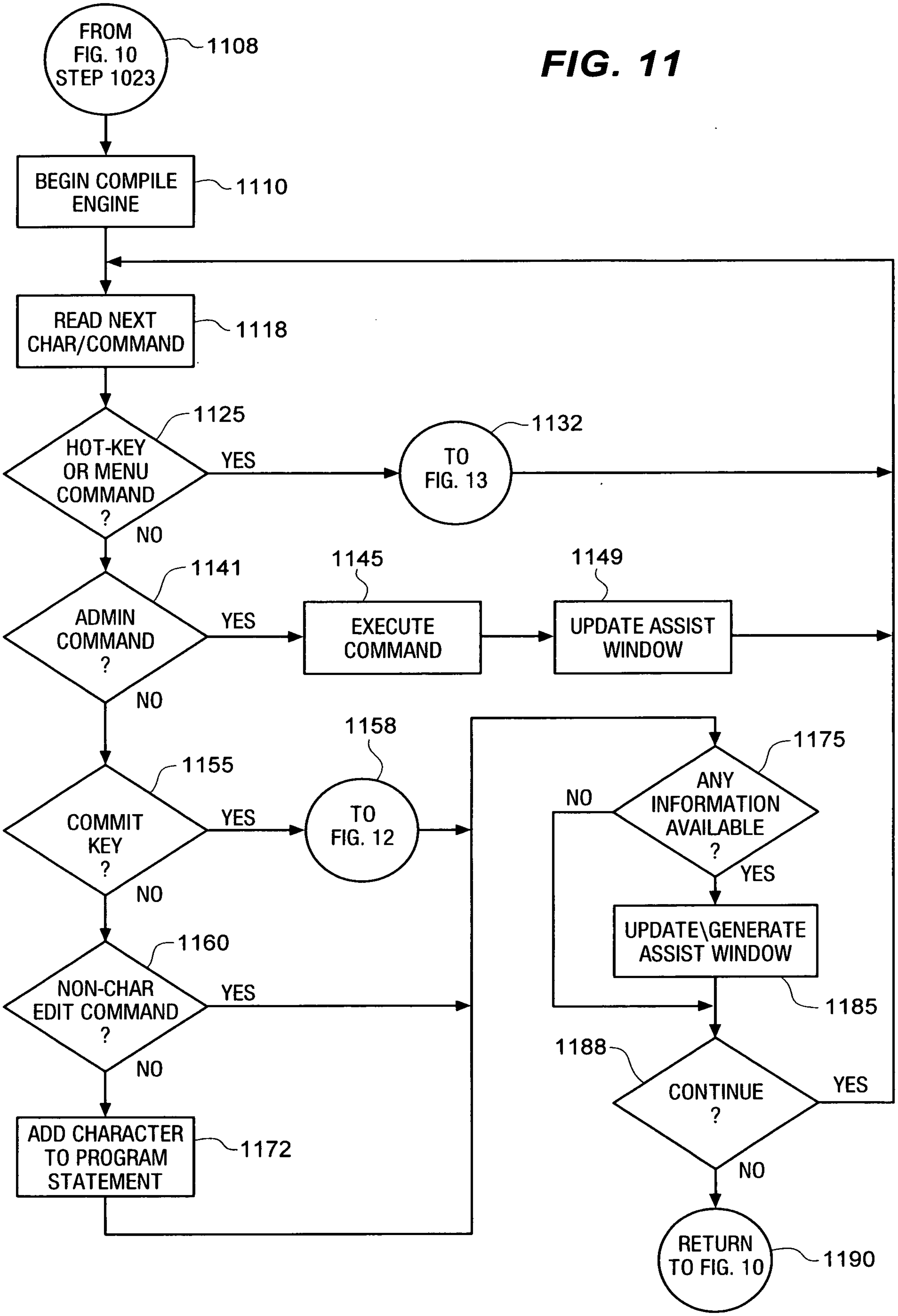

FROM FIG. 10 STEP 1023
1108
BEGIN COMPILE ENGINE
1110
READ NEXT CHAR/COMMAND
1118
HOT-KEY OR MENU COMMAND ?
1125
YES
NO
TO FIG. 13
1132
ADMIN COMMAND ?
1141
YES
NO
EXECUTE COMMAND
1145
UPDATE ASSIST WINDOW
1149
COMMIT KEY ?
1155
YES
NO
TO FIG. 12
1158
NON-CHAR EDIT COMMAND ?
1160
YES
NO
ADD CHARACTER TO PROGRAM STATEMENT
1172
ANY INFORMATION AVAILABLE ?
1175
NO
YES
UPDATE\GENERATE ASSIST WINDOW
1185
CONTINUE ?
1188
YES
NO
RETURN TO FIG. 10
1190

FIG. 12

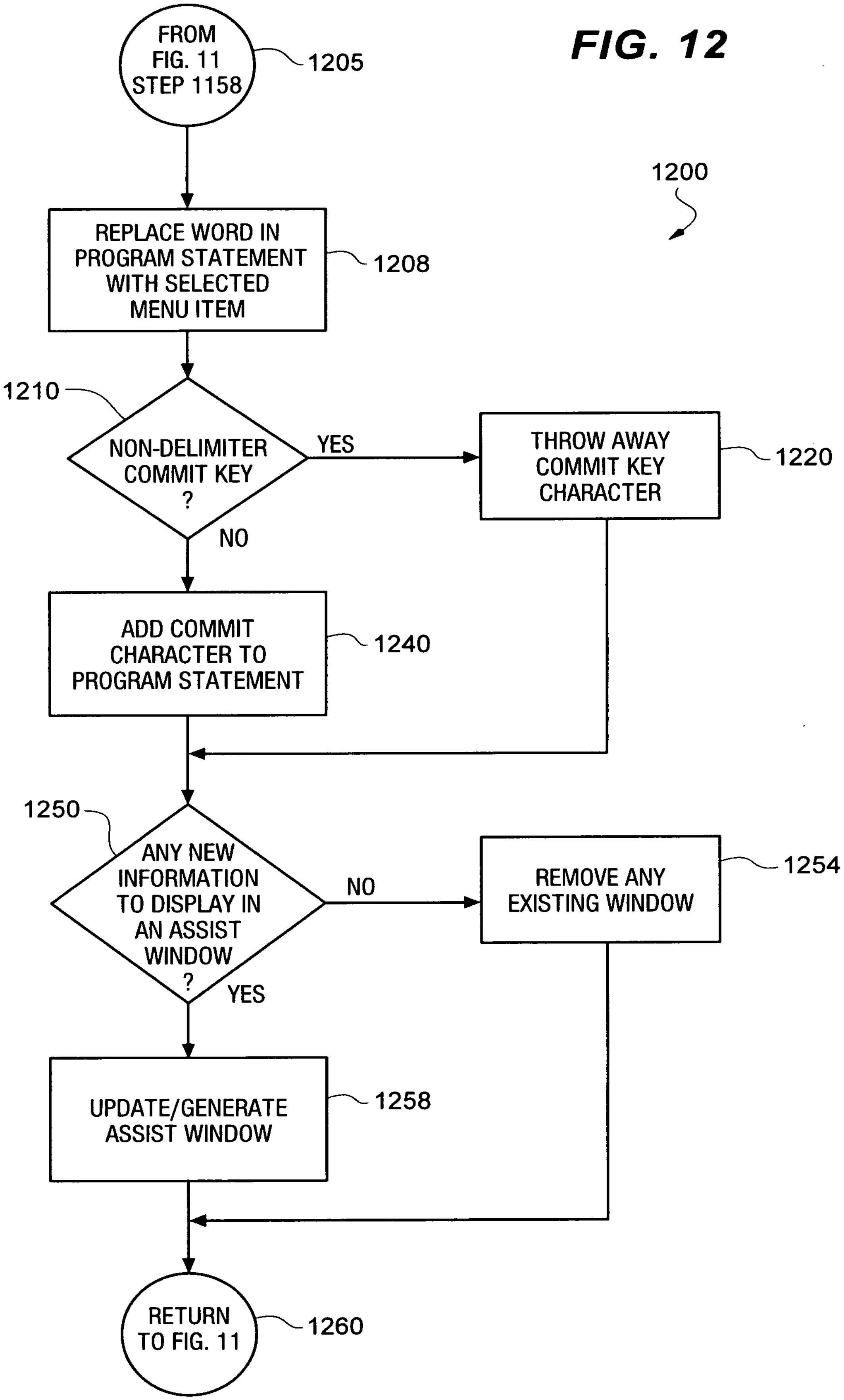
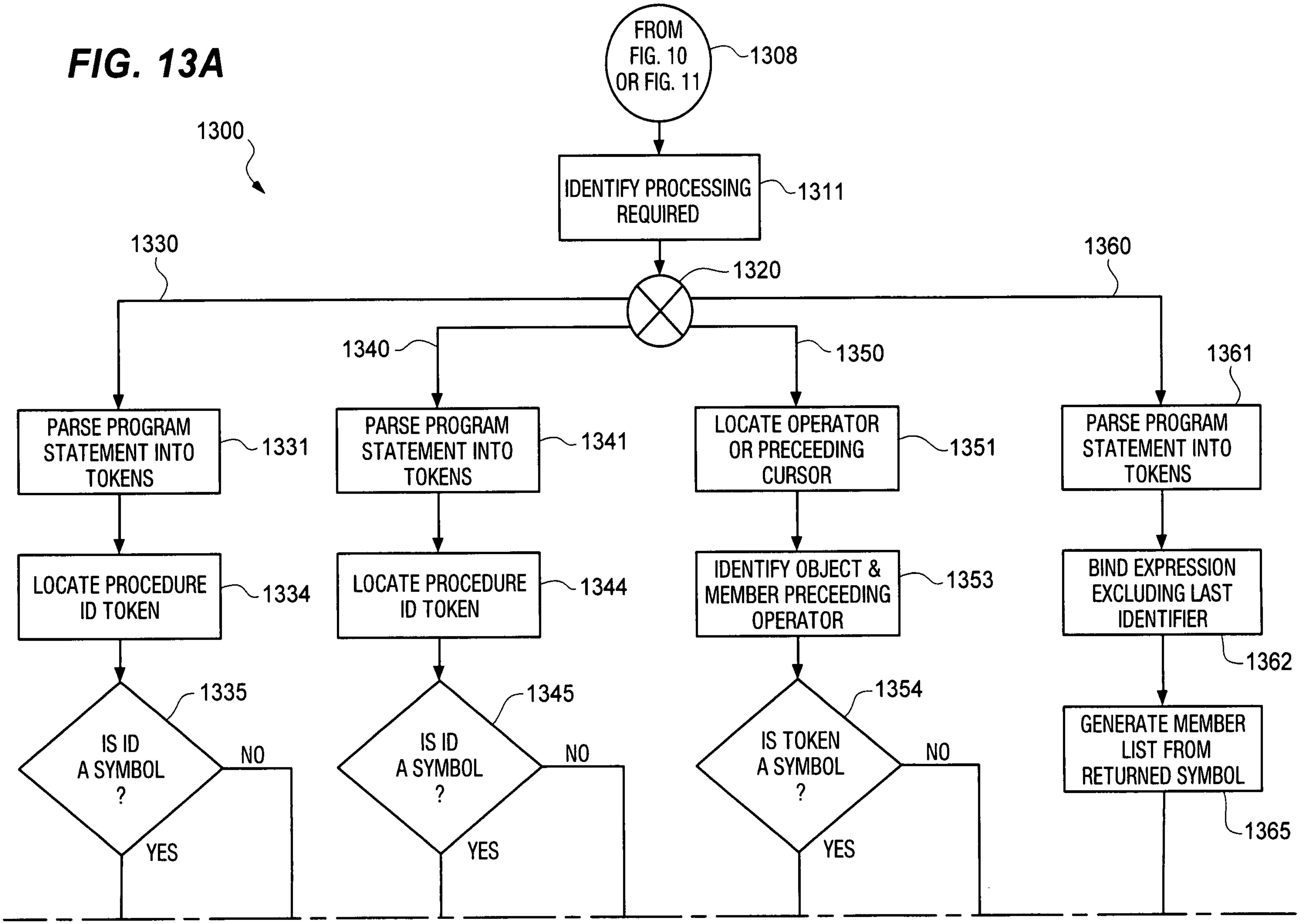

1200
FROM FIG. 11 STEP 1158
1205
REPLACE WORD IN PROGRAM STATEMENT WITH SELECTED MENU ITEM
1208
1210
NON-DELIMITER COMMIT KEY ?
YES
NO
THROW AWAY COMMIT KEY CHARACTER
1220
ADD COMMIT CHARACTER TO PROGRAM STATEMENT
1240
1250
ANY NEW INFORMATION TO DISPLAY IN AN ASSIST WINDOW ?
NO
YES
REMOVE ANY EXISTING WINDOW
1254
UPDATE/GENERATE ASSIST WINDOW
1258
RETURN TO FIG. 11
1260

FIG. 13A


1300
FROM FIG. 10 OR FIG. 11
1308
IDENTIFY PROCESSING REQUIRED
1311
1320
1330
1340
1350
1360
PARSE PROGRAM STATEMENT INTO TOKENS
1331
LOCATE PROCEDURE ID TOKEN
1334
IS ID A SYMBOL ?
1335
NO
YES
PARSE PROGRAM STATEMENT INTO TOKENS
1341
LOCATE PROCEDURE ID TOKEN
1344
IS ID A SYMBOL ?
1345
NO
YES
LOCATE OPERATOR OR PRECEEDING CURSOR
1351
IDENTIFY OBJECT & MEMBER PRECEEDING OPERATOR
1353
IS TOKEN A SYMBOL ?
1354
NO
YES
1361
PARSE PROGRAM STATEMENT INTO TOKENS
BIND EXPRESSION EXCLUDING LAST IDENTIFIER
1362
GENERATE MEMBER LIST FROM RETURNED SYMBOL
1365

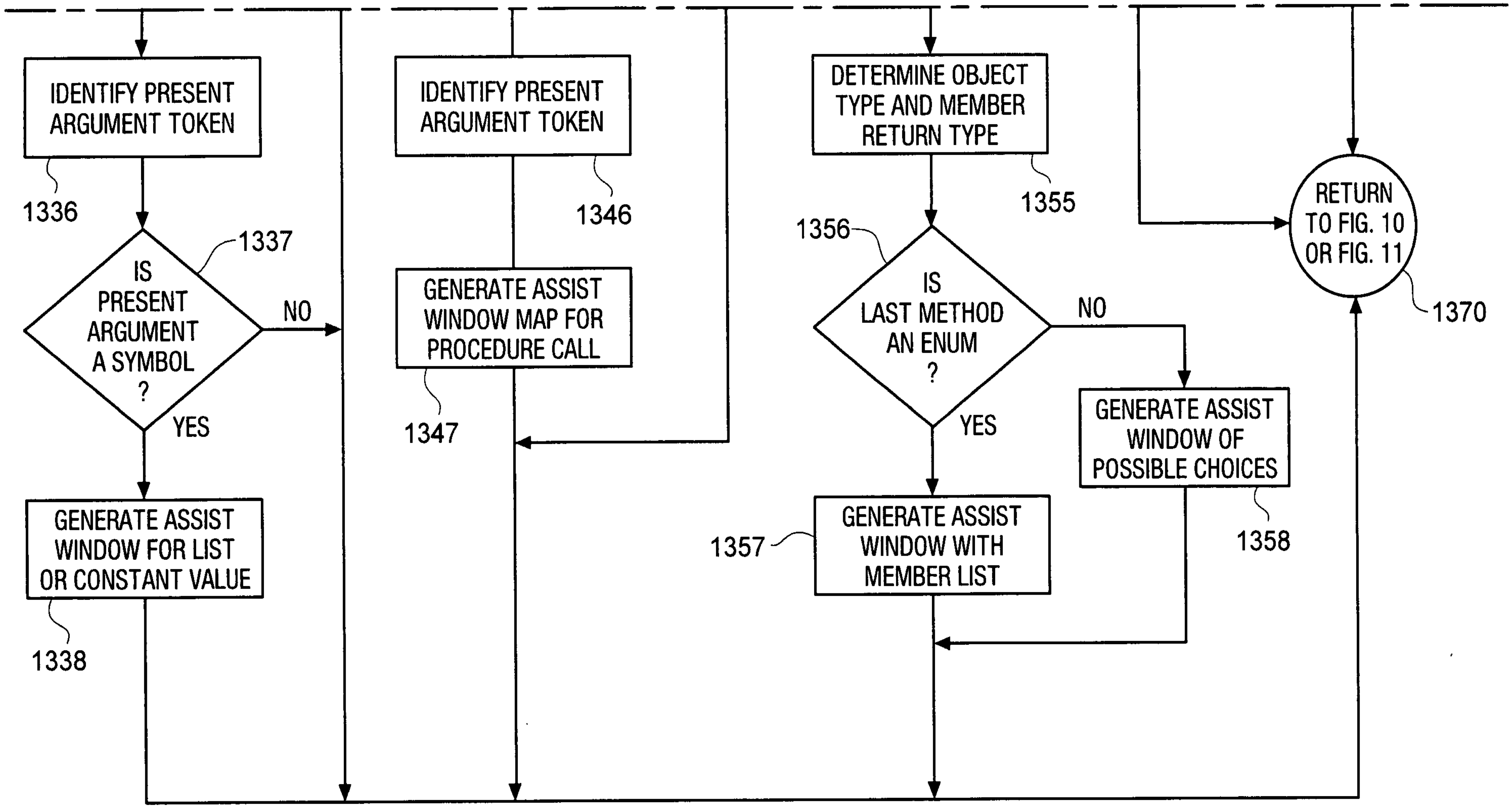

IDENTIFY PRESENT ARGUMENT TOKEN
1336
IS PRESENT ARGUMENT A SYMBOL ?
1337
NO
YES
GENERATE ASSIST WINDOW FOR LIST OR CONSTANT VALUE
1338
IDENTIFY PRESENT ARGUMENT TOKEN
1346
GENERATE ASSIST WINDOW MAP FOR PROCEDURE CALL
1347
DETERMINE OBJECT TYPE AND MEMBER RETURN TYPE
1355
1356
IS LAST METHOD AN ENUM ?
NO
YES
GENERATE ASSIST WINDOW OF POSSIBLE CHOICES
1358
1357
GENERATE ASSIST WINDOW WITH MEMBER LIST
RETURN TO FIG. 10 OR FIG. 11
1370

FIG. 13B